

Exercícios

Exercício 1: Carrinho de compras e pagamento

1. Criação de um carrinho de compras genérico:

- Implemente uma classe `Carrinho<T>` que permita armazenar itens genéricos.
- Deve conter métodos para adicionar e remover itens do carrinho.
- Deve haver um método que retorne a quantidade total de itens.

Dica: Utilize uma `List<T>` para armazenar os itens do carrinho.

Exercício 1: Carrinho de compras e pagamento

2. Sistema de pagamento genérico:

- Crie uma interface `Pagamento<T>` com um método `processarPagamento(T valor)`.
- Implemente duas classes `PagamentoCartao` e `PagamentoPix` que aceitem valores diferentes.
- Simule uma transação processando um pagamento com cada método.

Dica: Utilize **`T extends Number`** para garantir que o valor seja numérico.

Exercício 1: Carrinho de compras e pagamento

3. Gestão de pedidos genérica:

- Implemente uma classe `Pedido<K, V>` que armazene um identificador do pedido e uma lista de itens.
- O identificador deve ser genérico (K), podendo ser um número ou código alfanumérico.
- A lista de itens deve ser um conjunto de objetos (V).

Dica: Utilize um `Map<K, List<V>>` para armazenar os pedidos.

Exercício 2: Classe genérica para diferentes tipos de cupons de desconto

Cenário: O e-commerce precisa de uma funcionalidade para aplicar cupons de desconto a diferentes categorias de produtos.

Requisitos:

- Crie uma classe `CupomDesconto<T extends Number>`, onde `T` pode ser um `Integer` (para descontos fixos) ou `Double` (para percentuais).
- Adicione um método `aplicarDesconto(T desconto, double precoOriginal)` que retorne o preço final após o desconto.

Obrigada